A METHOD AND APPARATUS FOR UPDATING A HIERARCHICAL Z BUFFER

TECHNICAL FIELD OF THE INVENTION

The present invention relates to a graphics system of a computer graphics display system and, more particularly, to a method and apparatus for updating a hierarchical Z buffer in a graphics system of a computer graphics display system.

BACKGROUND OF THE INVENTION

In 3-D graphics systems, primitives are defined by X, Y and Z coordinates of the primitive, by the normals of the vertices of the primitive, and by the colors of the vertices. A graphics system has been proposed that would utilize a Z pyramid memory element which stores the Z values for a current frame of pixels to test for primitive occlusion. Such a graphics system was proposed in an article entitled "Hierarchical Z-Buffer Visibility", SIGGRAPH 93, Anaheim, California, 1-6 August 1993, by Greene, *et al.* This article discusses building a Z pyramid from the original Z buffer and using the Z pyramid to test cubes comprised of a plurality of primitives for occlusion culling. This approach makes use of the observation that the cube is hidden with respect to the Z buffer if all of its faces correspond to hidden polygons.

Prior to performing this comparison, the Z pyramid is constructed. The Z pyramid is constructed by starting with pixel-level Z values as the base of the pyramid and by comparing subgroups of adjacent Z values to determine the maximum Z value for each subgroup. The maximum Z value for that subgroup is then stored in the Z buffer for the next level up in the pyramid. For example, if 64 Z values were contained in the Z buffer when the comparison began, the next level up in the pyramid will contain 16 Z values, with each of the 16 Z values corresponding to the maximum Z value of the four pixels compared in the corresponding subgroup in the level below it.

In this next level up, the Z values of each subgroup of four adjacent Z values are compared to determine the maximum Z value for each subgroup and

then this maximum Z value is stored in the Z buffer for the next level up in the pyramid. Therefore, for the next level up in the pyramid, the Z buffer will contain four Z values. These four Z values are then compared to determine the maximum Z value, which is stored in the Z buffer for the next level up in the pyramid. This last maximum Z value corresponds to the top of the pyramid.

Once the Z pyramid has been constructed, the Z values for the primitives making up the faces of the cubes are compared beginning at the highest level of the pyramid and working down towards the base of the pyramid. For each primitive, the minimum Z value of the primitive is tested against the levels of the pyramid in a top-to-bottom sequence. If the minimum Z value for the primitive is greater than the pyramid Z value, then the primitive is fully occluded. If all of the primitives making up the faces of the cube are occluded, the cube is occluded and can be discarded. If not, the process continues at each level of the pyramid until the cube is either fully occluded or until the process reaches the level in the Z pyramid at which a primitive is found to be at least partially visible.

Although this approach is capable of simultaneously testing a plurality of primitives instead of individually testing each primitive, this approach does not make any provision for updating the Z pyramid "on the fly". Once a determination is made that a cube contains a visible primitive, the visible primitive must be scan converted to the pixel level before the Z buffer can be updated and the Z pyramid can be reconstructed.

Accordingly, a need exists for a method and apparatus for updating the Z buffer and reconstructing the Z pyramid on the fly as primitives are processed through the graphics pipeline.

SUMMARY OF THE INVENTION

The present invention provides a method and apparatus for occlusion testing primitives being processed in a graphics system and for updating a Z pyramid data structure used for occlusion testing on the fly. The apparatus comprises logic configured to create the Z pyramid data structure and to perform occlusion testing. The Z pyramid data structure comprises a plurality of levels,

each of which comprises a plurality of regions. Each region comprises a plurality of subregions, each of which corresponds to a single Z value. Each region corresponds to a plurality of Z values and has a maximum region Z value which corresponds to the largest Z value of the region. The logic compares the minimum Z value of each primitive with the Z value of a region associated with the primitive to determine whether or not the primitive is fully occluded.

If a determination is made that the primitive is not fully occluded, the logic determines whether or not any subregion of the region associated with the primitive is fully covered by the primitive. If the logic determines that a subregion is fully covered by the primitive, then the logic determines whether or not the Z value of the covered subregion needs to be replaced with the maximum Z value of the tested primitive. In order for the logic to determine whether the Z value of the covered subregion needs to be replaced with the maximum Z value of the primitive, the logic determines whether or not the maximum Z value of the primitive is less than the Z value for the covered subregion.

If the logic determines that the maximum Z value of the primitive is less than the Z value for the covered subregion, then the Z value for the covered subregion is replaced with the maximum Z value of the primitive. Preferably, the logic maintains a coverage mask for each level of the Z pyramid data structure. Each coverage mask comprises a bit for each subregion of the level of the Z pyramid data structure associated with the coverage mask. When the logic determines that the maximum Z value of the primitive is less than the Z value for the covered subregion, a bit in the coverage mask associated with the covered subregion is set. When all of the coverage mask bits corresponding to the subregions of a particular region have been set, a bit is set for the corresponding region in the coverage mask associated with the next level up in the Z pyramid data structure.

When all of the bits in the coverage mask have been set for a particular region, the logic replaces the maximum Z value for the particular region with the maximum Z value of all of the subregions associated with the particular region. When all of the bits in the coverage mask have been set for a particular region in

the coverage mask, the logic sets the corresponding bit in the coverage mask for the next level up in the Z pyramid.

Preferably, the primitives are occlusion tested in a tiler component of the graphics system and the Z pyramid data structure is updated by the tiler component on the fly as primitives are being processed through the graphics system. The graphics system is comprised as part of a computer graphics display system. The Z pyramid data structure preferably is stores in a Z pyramid memory element which is in communication with the tiler. Preferably, the Z pyramid memory element is periodically updated with pixel level Z values, i.e., with Z values of primitives which have been scan converted into screen coordinates corresponding to locations on the display monitor. In this way, the Z pyramid data structure can be updated on the fly and can be periodically updated with pixel level Z values to ensure accurate occlusion testing.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a computer graphics display system incorporating the graphics system of the present invention.

Fig. 2 is a block diagram of the graphics system of the computer graphics display system shown in Fig. 1.

Fig. 3 is a block diagram of the macro-function unit of the graphics system shown in Fig. 2 in accordance with the preferred embodiment of the present invention.

Figs. 4A-4D illustrate different levels of the Z pyramid used for occlusion testing by the macro-function unit shown in Fig. 3.

Figs. 5A-5C correspond to coverage masks used for updating the Z pyramid levels illustrated in Figs. 4A-4C.

Figs. 6 - 8 are flow charts illustrating the method of the present invention for performing occlusion testing and for updating the Z pyramid on the fly.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 1 is a block diagram of a computer graphics display system 10 comprising the graphics system 20 of the present invention in accordance with the preferred embodiment. The computer graphics display system 10 comprises a host CPU 12, a host memory device 14, a local bus 18, an input/output (I/O) controller device 25, an I/O bus 16, a graphics system 20, and a monitor 21 for displaying graphics information output from the graphics system 20.

The host CPU 12 processes input received from the console (not shown) of the computer graphics display system 10 and outputs commands and data over the local bus 18 to the I/O interface controller 25. The I/O interface controller 25 formats the commands and data utilizing the protocols of the I/O bus 16. The information received over the I/O bus 16 is input to the graphics system 20. The graphics system 20 then processes this information and causes graphics images to be displayed on the monitor 21.

Fig. 2 is a block diagram of the graphics system 20 of the present invention in accordance with the preferred embodiment. The host interface unit 32 fetches command data packets and texture maps from the host memory 14 via the I/O bus 16. The host interface unit 32 then provides graphics 2D information to the 2D macro-function unit (MFU) 34 and 3D information to the 3D macro-function unit (MFU) 36. The 2D macro-function unit 34 generates 2D vectors, text and rectangle spans. The 3D macro-function unit 36 performs triangle setup, 3D rastorization, and texture mapping, and pixel-level Z comparison operations, as discussed in more detail below with respect to Fig. 3. The 3D MFU 36 also performs occlusion testing in accordance with the present invention, as discussed in detail below with respect to Figs. 3-7.

The output from the 2D and 3D macro-function units 34 and 36 is received by the object function unit (OFU) 38. The object function unit 38 performs rectangle clipping, patterning, frame buffer-to-frame buffer block transfers and rectangle span fills. The output of the object function unit 38 is received by the frame buffer controller (FBC) 40. The frame buffer controller 40 dispatches requests to the memory controller (MC) 41 to cause the memory

controller 41 to write and read pixel colors and Z coordinates to and from RAM 45. The frame buffer controller 40 also fetches display information which is sent to the display controller 43. The display controller 43 receives the display information and converts it into red, green and blue (RGB) analog data and sends it to the display monitor 21.

Fig. 3 is a block diagram of the 3D macro-function unit 36 shown in Fig. 2. The setup unit 55 of the 3D macro-function unit 36 sets up edge equations for the polygons by using the X, Y values for the vertices of the polygons. The Z values of the vertices are used by the setup unit 55 to determine the minimum and maximum Z values for each polygon, as discussed below in detail with respect to Figs. 4A-8. The minimum and maximum Z values for each polygon are then output to the hierarchical tiler 57. The hierarchical tiler 57 builds the Z pyramid on the fly, i.e., as the polygons are being processed through the graphics pipeline. The hierarchical tiler 57 performs occlusion testing using the minimum and maximum Z values for the polygons received from the setup unit 55. The hierarchical tiler 57 also generates and maintains the coverage masks in accordance with the present invention to update the Z pyramid, as discussed below in detail with respect to Figs. 4A-8.

The hierarchical tiler 57 generates X and Y values corresponding to the minimum and maximum Z values for each of the visible primitives and provides the X and Y values to the parameter interpolator 58. The parameter interpolator 58 then calculates the exact Z values at the pixel level for each of the X and Y values received.

The exact Z values are then provided to the pixel Z-compare component 61. The pixel Z-compare component 61 then determines, at the pixel level, which values need to be written by the memory controller 41 of the frame buffer controller 40 to the Z buffer memory contained in RAM 45. In order to perform this pixel-level Z comparison test, the 3D macro-function unit 36 preferably comprises a pixel-level Z buffer memory 66 which stores the Z values obtained from the Z buffer memory comprised in RAM 45. The pixel Z compare component 61 uses these Z values to perform the pixel-level Z comparison tests.

Generally, the pixel-level Z comparison test compares batches of Z values stored in the frame buffer memory with the Z values generated by the parameter interpolator 58 to determine whether the Z values generated by the parameter interpolator 58 need to be stored in the Z buffer memory, or whether the pixel corresponding to the Z value is occluded. The pixel-level Z values to be stored in the Z buffer memory comprised in RAM 45 are then output to the object function unit 38 which then performs the tasks of rectangle clipping, patterning, frame buffer-to-frame buffer block transfers, rectangle span fills, etc.

Figs. 4A-4D illustrate various levels of the Z pyramid date structure stored in Z pyramid memory element 63 and used for occlusion testing. Fig. 4A represents the base level of the Z pyramid and Figs. 4B-4D represent successively higher levels of the pyramid, with Fig. 4D representing the top level of the pyramid. For ease of illustration and discussion, the base level of the Z pyramid is represented in Fig. 4A as containing 64 Z values 73 which are grouped into 16 subregions of adjacent Z values, each having four Z values. Each four adjacent Z values of each subregion 76 are compared to determine the maximum Z value for each subregion. The maximum Z values for each of the subregions are shown in Fig. 4B and represented by numeral 81. For this level of the pyramid 79, four subregions exist. Each subregion 83 in this level comprises four Z values. The four Z values for each of the four subregions 83 in level 79 are compared and the maximum Z values for each of the subregions 83 are saved. Fig. 4C represents the four maximum Z values resulting from the comparison of the Z values of the subregions 83 shown in Fig. 4B. The four Z values 87 of level 85 shown in Fig. 4C are then compared and the maximum Z value corresponds to the top level of the pyramid 93 shown in Fig. 4D.

In accordance with the present invention, coverage masks are used to indicate whether or not regions and/or subregions are covered by a polygon. These coverage masks are used to determine whether the Z pyramid needs to be updated and, if so, which Z values of the pyramid need to be updated. Figs. 5A-5C illustrate the coverage masks associated with the different levels of the pyramid represented by Figs. 4A-4C, respectively. These coverage masks enable the Z

pyramid to be updated on the fly as polygons are being processed through the pipeline of the graphics system 20. The coverage mask for each level of the pyramid preferably comprises one bit for each Z value in the corresponding level of the pyramid. Each of these individual Z values correspond to a subregion within the level of the pyramid associated with the particular mask. For example, level 71 of the pyramid shown in Fig. 4A comprises 16 regions, each of which comprises 4 subregions. Therefore, the corresponding coverage mask shown in Fig. 5A comprises 64 entries, i.e., one bit per subregion.

For each level of the pyramid, a Z MAX. PYR. value is stored for each region, which corresponds to the maximum Z value for the region. This value is used for region-by-region occlusion testing. A Z value is also stored for each subregion. This value is only used for occlusion testing if occlusion testing has resulted in a prior determination that the polygon is not fully occluded and does not fully cover the region. However, the values for the individual subregions are used for updating the Z pyramid, as described in detail below with respect to Fig. 7.

If the minimum Z value for the polygon, Z MIN. POLY., is less than the Z value for the subregion and the polygon fully covers the region, then the Z value for the subregion is updated with the maximum Z value for the polygon, Z MAX. POLY. However, the Z value for the subregion is only updated if the Z MAX. POLY. value is less than the current Z value for the subregion, Z MAX. PYR. The corresponding bit for the subregion is then set in the coverage mask corresponding to that level of the pyramid. When all of the bits of a subregion are set, the corresponding region bit is set in the coverage mask associated with the next level up in the Z pyramid. For example, with respect to pyramid level 71 shown in Fig. 4A, if subregion 73 and the three subregions adjacent to it are all covered by a polygon, the corresponding bits at locations 101, 102, 103 and 104 in coverage mask 100 shown in Fig. 5A will be set. Consequently, the corresponding bit 115 in the coverage mask 113 for the next level up in the pyramid shown in Fig. 5B will also be set.

When some but not all of the bits for a region are set, as indicated by locations 107, 108, 109 and 110 in coverage mask 100 shown in Fig. 5A, the corresponding bit in the coverage mask for the next level up in the pyramid will not be set, as indicated in coverage mask 113 shown in Fig. 5B. Since only one of the bits in the coverage mask 113 shown in Fig. 5B are set, none of the bits in the coverage mask 120 shown in Fig. 5C are set.

When all of the bits in the coverage mask associated with a region are set, the Z values for all of the subregions of the region are compared to find the maximum value. This maximum Z value becomes the Z MAX. PYR. value for that region and is used to perform occlusion testing.

The method by which the Z pyramid represented by Figs. 4A-4D is used to perform occlusion testing will now be discussed with respect to the flow charts of Figs. 6-8. In accordance with the preferred embodiment of the present invention, the polygons are occlusion tested on a region-by-region basis by comparing the minimum Z value of each polygon against the maximum Z value for each region in the current level of the pyramid. A subregion-by-subregion comparison preferably is only performed if the region is not fully occluded by the polygon and if the region is not fully covered by the polygon.

Assuming that the Z pyramid has already been constructed, the first block of the occlusion testing is to obtain the minimum Z value, Z MIN. POLY. for the polygon being occlusion tested, as indicated by block 122. The Z MIN. POLY. value is then compared to the Z MAX. PYR. value for the regions of the current level of the pyramid that intersect with the polygon, as indicated by block 124. Starting with the top level of the pyramid, a determination is made as to whether the Z MIN. POLY. value is less than the Z MAX. PYR. value, as indicated by block 125. If so, the polygon is at least partially visible, and the process proceeds to block 140 in Fig. 7, as indicated by block 127, for occlusion testing on a subregion-by-subregion basis, as described below in detail.

If a determination is made at block 125 that the Z MIN. POLY. value is not less than the Z MAX. PYR. value, the process proceeds to block 129 and a

determination is then made as to whether there are anymore regions at the current level of the pyramid to be tested, as indicated by block 129.

When a determination is made at block 125 that the Z MIN. POLY. value is less than the Z MAX. PYR. value for the region, which means that the polygon being tested is at least partially visible, the process proceeds to block 140 in Fig. 7. The Z value for each subregion is then compared with the Z MIN. POLY. value to determine whether or not the polygon is occluded by the subregions. If a determination is made at block 141 that the Z MIN. POLY. value is not less than or equal to the Z value for the subregion, a determination is made at block 142 as to whether there are any more subregions to be compared within the region. If not, the process returns to block 129 in Fig. 6. If, at block 142, a determination is made that there are more subregions within the region to be compared, the process proceeds to the next subregion within the region, as indicated by block 143. The Z value for the next subregion is then compared with the Z MIN. POLY. value, as indicated by block 140.

If a determination is made at block 141 that the Z MIN. POLY. value is less than or equal to the Z value for the subregion being compared, the process proceeds to block 144 where a determination is made as to whether the subregion is fully covered by the polygon. If so, a determination is then made as to whether the maximum Z value for the polygon being tested, Z MAX. POLY., is less than the Z MAX. PYR. value for the subregion, as indicated at block 145. If it is, the Z MAX. PYR. value for the subregion is assigned the value of the maximum Z value for the polygon, Z MAX. POLY., as indicated at block 146. The corresponding bit for the subregion is then set in the region coverage mask, as indicated at block 147, and then the process proceeds to block 150 in Fig. 7, as discussed below in detail, to determine whether the Z pyramid needs to be reconstructed with the updated Z buffer values.

If a determination is made at block 144 that the polygon does not fully cover the subregion, the process proceeds to the next level down in the pyramid, as indicated by block 149, and then returns to block 140 so that the polygon can be tested against the subregions in this level of the pyramid.

When the process proceeds to block 150 in Fig. 8, a determination is made as to whether or not the bits for all of the subregions of the current region have been set in the coverage mask. If not, the process returns to block 142 in Fig. 7 where a determination is made as to whether there are any more subregions to be tested against the polygon.

If a determination is made at block 150 that all bits corresponding to the region have been set in the coverage mask, and that the process is not at the top level of the pyramid, the maximum Z value for the region, Z MAX. PYR., is then assigned a value equal to the maximum Z value of all of the subregions, as indicated by block 153. This new value is then used in the block 125 in Fig. 6 to perform occlusion testing of the polygons on a region-by-region basis. The bit in the coverage mask corresponding to the region is then set in the coverage mask for the next level up in the pyramid, as indicated by block 155. The process then returns to block 142 in Fig. 7. The process also returns to block 142 when a determination is made at block 150 that either all bits for the current region are not set or that the process has reached the top level of the pyramid.

Whenever a coverage mask associated with a particular level of the Z pyramid is updated, a determination must be made as to whether the coverage masks associated with the higher levels in the pyramid must also be updated. For example, when the bits for all of the subregions associated with a region have been set in a coverage mask, the coverage mask associated with the next level up in the pyramid will need to be set. The process then proceeds to the next level up in the pyramid and a determination is then made as to whether the coverage mask associated with that level up in the pyramid needs to be set. This process continues until either the top level of the pyramid is reached or a determination is made that the corresponding bits in the coverage mask do not need to be set. When less than all of the bits for the subregions associated with a region have been set in a coverage mask, the coverage mask associated with the next level up in the pyramid will not need to be set. By updating the coverage masks in this manner, the Z pyramid is updated on the fly as the polygons are being processed through the graphics system 20.

Since the Z pyramid is updated on the fly using the minimum and maximum Z values for the polygons, the Z pyramid preferably is periodically rebuilt using pixel-level Z values provided from the pixel-level Z buffer memory element 66 to the Z pyramid memory element 63. This ensures the accuracy of the Z values used for occlusion testing and of the pixel-level Z values output from the MFU 36 to the OFU 38.

It should be noted that the present invention has been described with reference to the preferred embodiments of the present invention and that the present invention is not limited to these embodiments. Those skilled in the art will understand that modifications may be made to the embodiments discussed above without deviating from the spirit and scope of the present invention.